

1        **In the Claims**

2        Claims 1, 5, 16, 28 and 29 are amended.

3        Claims 1, 3-16, 18-42 are pending and are listed below:

4  
5        1.        (Currently Amended) A software architecture embodied on one or  
6        more computer-readable storage media, the software architecture for a distributed  
7        computing system comprising:

8                an application configured to handle requests submitted by remote devices  
9        over a network; and

10                an application program interface organized into multiple root namespaces,  
11        the application program interface to present functions used by the application to  
12        access network and computing resources of the distributed computing system,  
13        wherein calls to the application program interface are handed to a common  
14        language runtime layer ~~that can translate Web~~ supporting applications written in  
15        one or more different languages and translated into an intermediate supported  
16        language, the application program interface comprising various types related to  
17        constructing user interfaces, wherein the types belong to a group assigned a group  
18        name associated with one of the root namespaces, and wherein each of the types  
19        is referenced by a hierarchical name comprising a top level identifier prefixed to  
20        the group name.

21  
22        2.        (Canceled).

23  
24        3.        (Original) A software architecture as recited in claim 1, wherein the  
25        distributed computing system comprises client devices and server devices that

1 handle requests from the client devices, the remote devices comprising at least  
2 one client device.

3  
4 4. (Original) A software architecture as recited in claim 1, wherein the  
5 distributed computing system comprises client devices and server devices that  
6 handle requests from the client devices, the remote devices comprising at least  
7 one server device that is configured as a Web server.

8  
9 5. (Currently Amended) An application program interface embodied  
10 on one or more ~~tangible~~ computer readable storage media, comprising: multiple  
11 types related to constructing user interfaces, ~~the~~ individual types being associated  
12 with one or more groups and being referenced by one or more hierarchical names,  
13 wherein each hierarchical name includes a top level identifier prefixed to a group  
14 name assigned to one of the one or more groups, the types comprising classes  
15 which represent managed heap allocated data that has reference assignment  
16 semantics, interfaces that define a contract that other types can implement,  
17 delegates that are object oriented function pointers, structures that represent static  
18 allocated data that has value assignment semantics and enumerations which are  
19 value types that represent named constants, wherein the application program  
20 interface is associated with a common language runtime layer ~~that can translate~~  
21 ~~Web-supporting~~ applications written in one or more of several different languages  
22 and translated into an intermediate language supported by the common runtime  
23 layer.

1           6.     (Original) An application program interface as recited in claim 5,  
2 wherein the classes comprise a forms class that represents a window or a dialog  
3 box that makes up an application's user interface.

4  
5           7.     (Original) An application program interface as recited in claim 6,  
6 wherein the forms class has multiple members comprising one or more of: public  
7 static properties, public static methods, public instance constructors, public  
8 instance methods, public instance properties, public instance events, protected  
9 instance properties, and protected instance methods.

10  
11           8.     (Original) An application program interface as recited in claim 5,  
12 wherein the type comprising the interfaces comprises a button control interface  
13 that allows a control to act like a button on a form.

14  
15           9.     (Original) An application program interface as recited in claim 5,  
16 wherein the type comprising the interfaces comprises a container control interface  
17 that provides functionality for a control to act as a parent for other controls.

18  
19           10.    (Original) An application program interface as recited in claim 5,  
20 wherein the type comprising the interfaces comprises an editing notification  
21 interface.

22  
23           11.    (Original) An application program interface as recited in claim 5,  
24 wherein the type comprising the interfaces comprises a data object interface that  
25 provides a format independent mechanism for transferring data.

1  
2 12. (Original) An application program interface as recited in claim 5,  
3 wherein the type comprising the interfaces comprises a feature support interface  
4 that specifies a standard interface for retrieving feature information from a current  
5 system.

6  
7 13. (Original) An application program interface as recited in claim 5,  
8 wherein the type comprising the interfaces comprises a message filter interface.

9  
10 14. (Original) An application program interface as recited in claim 5,  
11 wherein the type comprising the interfaces comprises a handle-exposing interface  
12 to expose handles.

13  
14 15. (Original) An application program interface as recited in claim 5,  
15 wherein the type comprising the interfaces comprises one or more of the  
16 following interfaces:

17 a button control interface that allows a control to act like a button on a  
18 form;

19 a container control interface that provides functionality for a control to act  
20 as a parent for other controls;

21 an editing notification interface;

22 a data object interface that provides a format independent mechanism for  
23 transferring data;

24 a feature support interface that specifies a standard interface for retrieving  
25 feature information from a current system;

1 a message filter interface; and

2 a handle-exposing interface to expose handles.

3  
4 16. (Currently Amended) A distributed computer software architecture  
5 embodied on one or more computer-readable storage media, the distributed  
6 computer software architecture comprising:

7 one or more applications configured to be executed on one or more  
8 computing devices, the applications handling requests submitted from remote  
9 computing devices;

10 a networking platform to support the one or more applications;

11 an application programming interface to interface the one or more  
12 applications with the networking platform, the application programming interface  
13 comprising various types related to constructing user interfaces, individual types  
14 being associated with one or more groups and being referenced by one or more  
15 hierarchical names, wherein each of the hierarchical names includes a top level  
16 identifier prefixed to a group name assigned to one of the one or more groups;  
17 and

18 a common language runtime layer ~~that can translate Web supporting~~  
19 applications written in one or more different languages and translated into an  
20 intermediate language supported by the common runtime layer.

21  
22 17. (Canceled).  
23  
24  
25

18. (Previously Presented) A distributed computer software architecture as recited in claim 42, wherein the classes comprises a forms class that represents a window or a dialog box that makes up an application's user interface.

19. (Original) A distributed computer software architecture as recited in claim 18, wherein the forms class has multiple members comprising one or more of: public static properties, public static methods, public instance constructors, public instance methods, public instance properties, public instance events, protected instance properties, and protected instance methods.

20. (Previously Presented) A distributed computer software architecture as recited in claim 42, wherein the type comprising the interfaces comprises a button control interface that allows a control to act like a button on a form.

21. (Previously Presented) A distributed computer software architecture as recited in claim 42, wherein the type comprising the interfaces comprises a container control interface that provides functionality for a control to act as a parent for other controls.

22. (Previously Presented) A distributed computer software architecture as recited in claim 42, wherein the type comprising the interfaces comprises an editing notification interface.

23. (Previously Presented) A distributed computer software architecture as recited in claim 42, wherein the type comprising the interfaces comprises a

1 data object interface that provides a format independent mechanism for  
2 transferring data.

3  
4 24. (Previously Presented) A distributed computer software architecture  
5 as recited in claim 42, wherein the type comprising the interfaces comprises a  
6 feature support interface that specifies a standard interface for retrieving feature  
7 information from a current system.

8  
9 25. (Previously Presented) A distributed computer software architecture  
10 as recited in claim 42, wherein the type comprising the interfaces comprises a  
11 message filter interface.

12  
13 26. (Previously Presented) A distributed computer software architecture  
14 as recited in claim 42, wherein the type comprising the interfaces comprises a  
15 handle-exposing interface to expose handles.

16  
17 27. (Previously Presented) A distributed computer software architecture  
18 as recited in claim 42, wherein the type comprising the interfaces comprises one  
19 or more of the following interfaces:

20 a button control interface that allows a control to act like a button on a  
21 form;

22 a container control interface that provides functionality for a control to act  
23 as a parent for other controls;

24 an editing notification interface;

1 a data object interface that provides a format independent mechanism for  
2 transferring data;

3 a feature support interface that specifies a standard interface for retrieving  
4 feature information from a current system;

5 a message filter interface; and

6 a handle-exposing interface to expose handles.

7  
8 28. (Currently Amended) A computer system including one or more  
9 microprocessors and one or more software programs, the one or more software  
10 programs utilizing an application program interface to request services from an  
11 operating system, the application program interface including separate commands  
12 to request services comprising services related to constructing user interfaces,  
13 wherein the application program interface groups API functions into multiple  
14 namespaces that define a collection of classes which represent managed heap  
15 allocated data that has reference assignment semantics, interfaces that define a  
16 contract that other types can implement, delegates that are object oriented  
17 function pointers, enumerations which are value types that represent named  
18 constants and structures that represent static allocated data that has value  
19 assignment semantics, the application program interface being associated with a  
20 common language runtime layer ~~that can translate Web~~ supporting applications  
21 written in one or more different languages and translated into an intermediate  
22 language supported by the common runtime layer.

23  
24 29. (Currently Amended) A method, comprising:  
25



1 managing network and computing resources for a distributed computing  
2 system; and

3 exposing a set of functions that enable developers to access the network  
4 and computing resources of the distributed computing system, the set of functions  
5 comprising functions to facilitate construction of user interfaces, wherein the  
6 functions are grouped into multiple namespaces that define a collection of classes  
7 which represent managed heap allocated data that has reference assignment  
8 semantics, interfaces that define a contract that other types can implement,  
9 delegates that are object oriented function pointers, enumerations which are value  
10 types that represent named constants and structures that represent static allocated  
11 data that has value assignment semantics; and

12 using a common language runtime layer ~~that can translate Web~~ supporting  
13 applications written in one or more different languages and translated into an  
14 intermediate language supported by the common runtime layer.

15  
16 30. (Original) A method as recited in claim 29, further comprising  
17 receiving a request from a remote computing device, the request containing a call  
18 to the set of functions.

19  
20 31. (Previously Presented) A method, comprising creating a namespace  
21 with functions that enable drawing and construction of user interfaces, the  
22 namespace defining classes which represent managed heap allocated data that has  
23 reference assignment semantics, interfaces that define a contract that other types  
24 can implement, delegates that are object oriented function pointers, structures that  
25

1 represent static allocated data that has value assignment semantics, and  
2 enumerations which are value types that represent named constants.

3  
4 32. (Original) A method as recited in claim 31, wherein the namespace  
5 defines a forms class that represents a window or a dialog box that makes up an  
6 application's user interface.

7  
8 33. (Original) A method as recited in claim 32, wherein the forms class  
9 has multiple members comprising one or more of: public static properties, public  
10 static methods, public instance constructors, public instance methods, public  
11 instance properties, public instance events, protected instance properties, and  
12 protected instance methods.

13  
14 34. (Original) A method as recited in claim 31, wherein the namespace  
15 defines an interface comprising a button control interface that allows a control to  
16 act like a button on a form.

17  
18 35. (Original) A method as recited in claim 31, wherein the namespace  
19 defines an interface comprising a container control interface that provides  
20 functionality for a control to act as a parent for other controls.

21  
22 36. (Original) A method as recited in claim 31, wherein the namespace  
23 defines an interface comprising an editing notification interface.

1           37. (Original) A method as recited in claim 31, wherein the namespace  
2 defines an interface comprising a data object interface that provides a format  
3 independent mechanism for transferring data.  
4

5           38. (Original) A method as recited in claim 31, wherein the namespace  
6 defines an interface comprising a feature support interface that specifies a  
7 standard interface for retrieving feature information from a current system.  
8

9           39. (Original) A method as recited in claim 31, wherein the namespace  
10 defines an interface comprising a message filter interface.  
11

12           40. (Original) A method as recited in claim 31, wherein the namespace  
13 defines an interface comprising a handle-exposing interface to expose handles.  
14

15           41. (Previously Presented) A software architecture as recited in claim 1,  
16 wherein the various types comprise classes, interfaces, delegates, structures and  
17 enumerations.  
18

19           42. (Previously Presented) A distributed computer software architecture  
20 as recited in claim 16, wherein the various types comprise classes, interfaces,  
21 delegates, structures and enumerations.  
22  
23  
24  
25